Agnieszka Janowicz

Odległość Hamminga

Dokumentacja projektu MIPS

**Opis programu**

Program liczy minimalną odległość Hamminga między dwoma obrazami BMP, z uwzględnieniem możliwego przesunięcia w osi poziomej i pionowej obrazów względem siebie. Przesunięcie może zawierać się w przedziale <-7,7> pikseli.

Program tworzy dwa pliki tekstowe:

* hamming.txt – zawiera minimalną odległość Hamminga między obrazami dla dowolnego dopuszczalnego przesunięcia w obu osiach,
* tablica.txt – zawiera tabelę odległości Hamminga dla wszystkich możliwych przesunięć.

**Opis struktury programu**

Program dzieli się na 5 główne części:

1. Otwarcie plików i odczytanie danych.

2. Podwójna pętla, w której wykonywane są przesunięcia pionowe i poziome oraz obliczanie i zapisywanie do tablicy odległości Hamminga dla każdego przesunięcia.

3. Zapisanie odległości i zamknięcie plików wynikowych.

4. Funkcja itoa, zamieniająca liczbę całkowitą na kod ASCII.

5. Funkcja zamieniająca kolejność bajtów w rejestrze

**Opis struktur danych i ich implementacji**

W sekcji .data rezerwowana jest następująca przestrzeń:

* *buf1*, *buf2* – 512-bajtowe bufory przechowujące zawartości każdego obrazka. 1 bajt odpowiada 8 pikselom. Bajty są wczytywane w kolejności od ‘lewego dolnego rogu’ do ‘prawego górnego rogu’ obrazka.
* *temp* – 62-bajtowy bufor służący do wczytania nagłówków z plików .bmp
* *hamm* – 5-bajtowy bufor używany przez funkcję itoa do zapisu liczby w kodzie ASCII
* *odstep*, *odstep4*, *lin*, *linef* – bufory pomagające w tworzeniu pliku wynikowego tablica.txt
* *err1* – treść komunikatu błędu pojawiającego się przy błędnym odczycie pliku
* *outfile1*, *outfile2*, *infile1*, *infile2* – bufory przechowujące nazwy plików wynikowych i plików, na których program operuje

**Stosowane algorytmy**

Wczytywanie danych do pamięci

Do bufora buf1 wczytywana jest zawartość pierwszego pliku, a do buf2 zawartość drugiego pliku (z pominięciem 62 bajtów nagłówka).

Następnie do rejestrów $t0, $t2 wczytywane są po 4 bajty z pierwszego pliku (dwie połówki jednego wiersza), a do $t1 i $t3 po 4 bajty z drugiego pliku. Kolejność bajtów w rejestrach jest zamieniana (z powodu odwrotnego wczytywania bajtów do rejestru – *problem* *little endian – big endian*) poprzez zastosowania przesunięć logicznych w lewo i w prawo. Takie wczytanie jest wykonywane 64 razy dla każdego przesunięcia.

Przesunięcia obrazów w przedziale <-7,7> pikseli

Jeśli wartość przesunięcia jest większa od 0 to przesuwany jest obraz1 w granicach <0,7>, a jeśli mniejsza – przesuwany jest obraz2 w granicach (0,7> .Przy przesunięciach zakłada się, że część obrazka, która nie jest sprawdzana jest zgodna z drugim obrazkiem.

**Przesunięcie pionowe**

Dla wartości przesunięcia pionowego równej x indeks w bitmapie zostaje zwiększony 8\*x razy. Oznacza to, że przy dalszym wczytaniu danych do pamięci ominięte zostanie x pierwszych wierszy (liczonych od dołu, zgodnie ze sposobem wczytywania danych z bitmapy).

**Przesunięcie poziome**

Pierwsza połowa wiersza ($t0) jest przesuwana logicznie w lewo o x (wartość przesunięcia w poziomie), natomiast druga połowa ($t2) jest rotowana w lewo o x. Wtedy x pierwszych najmniej znaczących bitów w rejestrze $t0 jest równa 0, a w rejestrze $t2 zostaje zamieniona na x najbardziej znaczących bitów tego rejestru. Następnie jest liczona reszta z dzielenia drugiej połowy przez 2^$s0 (poprzez logiczny and tej liczby z 2x -1 ), w celu wyłuskania ‘przesuniętych’ bitów, tych, które po przesunięciu obrazka powinny znaleźć się w pierwszej połowie wiersza. Mamy wtedy część liczby, która została przeniesiona z najbardziej znaczących bitów na najmniej znaczące (znajduje się w rejestrze $t7). Od $t2 odejmuję $t7 (otrzymuję wtedy wartość taką jak po logicznym przesunięciu w lewo), a do $t0 dodaję $t7 (wtedy najmniej znaczące bity zajmują miejsce zer powstałych w $t0 na skutek przesunięcia logicznego).

Podobny algorytm stosuję dla $t1 i $t3.

Sprawdzanie zgodności pikseli

**Dla pierwszych połówek wiersza**

Zawartość $t0 i $t1 jest xorowana, wynik zostaje umieszczony w $s3. Następnie sprawdzam, czy liczba jest ujemna. Jeśli jest ujemna to znaczy, że najważniejszy bit liczby w rejestrze jest równy 1, a więc na tym bicie liczby zawarte w $t0 i $t1 się różnią. Wtedy do licznika różnic dodaję 1. Sprawdzanie takie jest wykonane 32 razy dla tych rejestrów.

**Dla drugich połówek wiersza:**

Dla $t2 i $t3 procedura jest podobna, ale sprawdzanie jest wykonane 32 - $s2 razy, gdzie $s2 – wartość bezwzględna przesunięcia w poziomie.

Zapisanie odległości Hamminga

Po każdym sprawdzeniu odległości Hamminga (przy każdych wartościach przesunięć) program zapisuje do pliku tablica.txt aktualną wartość Hamminga w odpowiednim miejscu w tabeli. Jednocześnie wartość ta jest porównywana z minimalną dotychczas znalezioną odległością (znajdującą się w rejestrze $s7 i inicjowaną liczbą 4096 – największą możliwą wartością). Jeśli aktualna wartość jest mniejsza, to jest kopiowana do $s7. Po wykonaniu wszystkich sprawdzeń program zapisuje minimalną odległość Hamminga do pliku hamming.txt.

Zasada działania głównej części programu

W zewnętrznej pętli następuje zapisanie poprzedniej wartości odległości Hamminga do tablicy (jeśli pętla nie jest wykonywana po raz pierwszy – w przeciwnym wypadku ta część jest omijana) oraz przesunięcie pionowe obrazka – to znaczy obliczenie indeksu początkowego w buforach buf1, buf2 zawierających porównywane obrazki. W pętli wewnętrznej wykonane jest wczytanie danych do pamięci oraz przesunięcie poziome. Następnie sprawdzana jest liczba różniących się bitów w rejestrach , a wynik dodawany do aktualnej odległości Hamminga. Dla każdej kombinacji przesunięć pętla wewnętrzna wykonuje się, dopóki indeks nie przekroczy rozmiaru bufora.

Przesunięcia są sprawdzane od wartości -7 do 7. Zewnętrzna pętla kończy się, gdy wartość przesunięcia pionowego przekroczy 7.

**Testy**

Wyniki przeprowadzonych testów zgadzały się z wynikami przewidywanymi. Poniżej prezentuję kilka z nich, testujących kilka granicznych przypadków.

1. Dla poniższych obrazków:

 

Obliczona minimalna odległość Hamminga (czyli zawartość pliku hamming.txt): 0

Tablica wszystkich wartości:

y\x | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

-7 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

-6 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

-5 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

-4 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

-3 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

-2 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

-1 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

0 35 35 35 35 35 35 35 35 28 21 14 7 0 0 0

1 30 30 30 30 30 30 30 30 24 18 12 6 0 0 0

2 25 25 25 25 25 25 25 25 20 15 10 5 0 0 0

3 20 20 20 20 20 20 20 20 16 12 8 4 0 0 0

4 15 15 15 15 15 15 15 15 12 9 6 3 0 0 0

5 10 10 10 10 10 10 10 10 8 6 4 2 0 0 0

6 5 5 5 5 5 5 5 5 4 3 2 1 0 0 0

1. 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2. Dla obrazków:



Obliczona minimalna odległość Hamminga: 0

Tablica wszystkich wartości:

y\x | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

-7 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

-6 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

-5 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

-4 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

-3 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

-2 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

-1 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

0 0 0 0 7 14 21 28 35 35 35 35 35 35 35 35

1 0 0 0 6 12 18 24 30 30 30 30 30 30 30 30

2 0 0 0 5 10 15 20 25 25 25 25 25 25 25 25

3 0 0 0 4 8 12 16 20 20 20 20 20 20 20 20

4 0 0 0 3 6 9 12 15 15 15 15 15 15 15 15

5 0 0 0 2 4 6 8 10 10 10 10 10 10 10 10

6 0 0 0 1 2 3 4 5 5 5 5 5 5 5 5

7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1. Dla obrazków:



Obliczona minimalna odległość Hamminga: 2

Tablica wszystkich wartości:

y\x | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

-7 24 24 24 24 24 16 24 24 24 24 24 24 24 24 24

-6 24 24 24 24 24 14 24 24 24 24 24 24 24 24 24

-5 24 24 24 24 24 12 24 24 24 24 24 24 24 24 24

-4 24 24 24 24 24 10 24 24 24 24 24 24 24 24 24

-3 24 24 24 24 24 8 24 24 24 24 24 24 24 24 24

-2 24 24 24 24 24 6 24 24 22 24 24 24 24 24 24

-1 24 24 24 24 24 4 24 24 22 24 24 24 24 24 24

0 24 24 24 24 24 2 22 24 22 24 24 24 24 24 24

1 24 24 24 24 24 4 22 24 22 24 24 24 24 24 24

2 24 24 24 24 24 6 22 24 22 24 24 24 24 24 24

3 24 24 24 24 24 8 22 24 22 24 24 24 24 24 24

4 24 24 24 24 24 10 22 24 22 24 24 24 24 24 24

5 24 24 24 24 24 12 22 24 22 24 24 24 24 24 24

6 24 24 24 24 24 14 22 24 22 24 24 24 24 24 24

7 24 24 24 24 24 16 22 24 22 24 24 24 24 24 24

1. Dla obrazków:

  (Identyczne obrazki – 1 czarny piksel w tym samym miejscu, reszta biała)

Obliczona minimalna odległość Hamminga: 0

Tablica wszystkich wartości:

y\x | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

-7 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

-6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

-5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

-4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

-3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

-2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

-1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

0 2 2 2 2 2 2 2 0 2 2 2 2 2 2 2

1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

5 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

7 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

1. Dla obrazków:

(Drugi obrazek jest kopią pierwszego przesuniętą w pionie i poziomie o (1,1) )

Obliczona minimalna odległość Hamminga: 0

Tablica wszystkich wartości:

y\x | -7 | -6 | -5 | -4 | -3 | -2 | -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

-7 98 98 98 98 98 98 98 98 98 98 98 98 98 98 98

-6 98 98 98 98 98 98 98 98 98 98 98 98 98 98 98

-5 98 98 98 98 96 94 92 94 94 94 92 94 96 98 98

-4 98 98 98 94 90 86 86 84 84 84 86 86 90 94 98

-3 98 98 94 88 82 78 74 72 70 70 72 76 82 86 92

-2 98 94 88 80 74 66 60 56 54 54 58 64 72 78 86

-1 96 90 82 72 64 54 46 40 36 38 44 54 64 72 80

0 94 86 76 66 58 46 34 24 18 22 34 46 58 66 76

1 92 84 74 64 52 44 30 16 0 16 30 44 52 64 74

2 92 84 76 66 58 46 34 22 18 24 34 46 58 66 76

3 94 88 80 72 64 54 44 38 36 40 46 54 64 72 82

4 96 92 86 78 72 64 58 54 54 56 60 66 74 80 88

5 98 96 92 86 82 76 72 70 70 72 74 78 82 88 94

6 98 98 98 94 90 86 86 84 84 84 86 86 90 94 98

7 98 98 98 98 96 94 92 94 94 94 92 94 96 98 98

1. Dla obrazków:

  (identyczne obrazki, wzór w paski po całej wysokości obrazka, łatwo przewidzieć rezultat)

Obliczona minimalna odległość Hamminga: 0

Tablica wszystkich wartości:

y\x | -7 | -6 | -5 | -4 | -3 | -2| -1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

-7 3249 0 3363 0 3477 0 3591 0 3591 0 3477 0 3363 0 3249

-6 3306 0 3422 0 3538 0 3654 0 3654 0 3538 0 3422 0 3306

-5 3363 0 3481 0 3599 0 3717 0 3717 0 3599 0 3481 0 3363

-4 3420 0 3540 0 3660 0 3780 0 3780 0 3660 0 3540 0 3420

-3 3477 0 3599 0 3721 0 3843 0 3843 0 3721 0 3599 0 3477

-2 3534 0 3658 0 3782 0 3906 0 3906 0 3782 0 3658 0 3534

-1 3591 0 3717 0 3843 0 3969 0 3969 0 3843 0 3717 0 3591

0 3648 0 3776 0 3904 0 4032 0 4032 0 3904 0 3776 0 3648

1 3591 0 3717 0 3843 0 3969 0 3969 0 3843 0 3717 0 3591

2 3534 0 3658 0 3782 0 3906 0 3906 0 3782 0 3658 0 3534

3 3477 0 3599 0 3721 0 3843 0 3843 0 3721 0 3599 0 3477

4 3420 0 3540 0 3660 0 3780 0 3780 0 3660 0 3540 0 3420

5 3363 0 3481 0 3599 0 3717 0 3717 0 3599 0 3481 0 3363

6 3306 0 3422 0 3538 0 3654 0 3654 0 3538 0 3422 0 3306

7 3249 0 3363 0 3477 0 3591 0 3591 0 3477 0 3363 0 3249

Przeprowadziłam ogólnie więcej testów, ale na podstawie przedstawionych powyżej można stwierdzić, że program działa poprawnie.